ENV 790.30 - Time Series Analysis for Energy Data | Spring 2023

Assignment 7 - Due date 03/20/23

Yuxiang Ren

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github. And to do so you will need to fork our repository and link it to your RStudio.

Once you have the file open on your local machine the first thing you will do is rename the file such that it includes your first and last name (e.g., "LuanaLima_TSA_A07_Sp23.Rmd"). Then change "Student Name" on line 4 with your name.

Then you will start working through the assignment by **creating code and output** that answer each question. Be sure to use this assignment document. Your report should contain the answer to each question and any plots/tables you obtained (when applicable).

When you have completed the assignment, **Knit** the text and code into a single PDF file. Submit this pdf using Sakai.

Set up

```
#Load/install required package here
#install.packages("uroot")
library(uroot)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tseries)
library(readr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(Kendall)
```

Importing and processing the data set

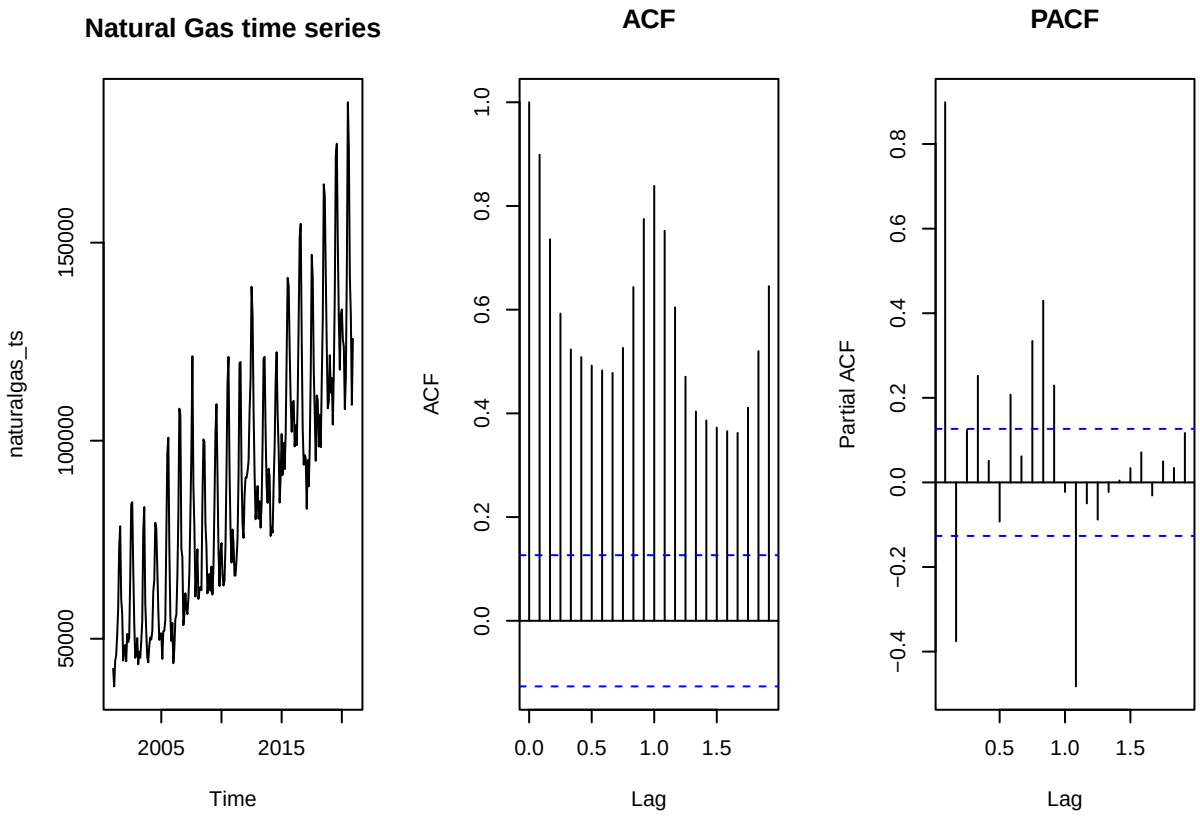
Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

Packages needed for this assignment: "forecast","tseries". Do not forget to load them before running your script, since they are NOT default packages.\

Q1

Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

```
rawdata <- read.csv( "./Data/Net_generation_United_States_all_sectors_monthly.csv",skip=4)
rawdata$Month <- dmy(paste0("01 ", rawdata$Month))
rawdata <- arrange(rawdata, Month)
naturalgas_ts <- ts(rawdata[,4], start = c(2001,1), frequency = 12)
par(mfrow = c(1,3))
plot(naturalgas_ts, main = "Natural Gas time series")
acf(naturalgas_ts, main = "ACF")
pacf(naturalgas_ts, main = "PACF")
```

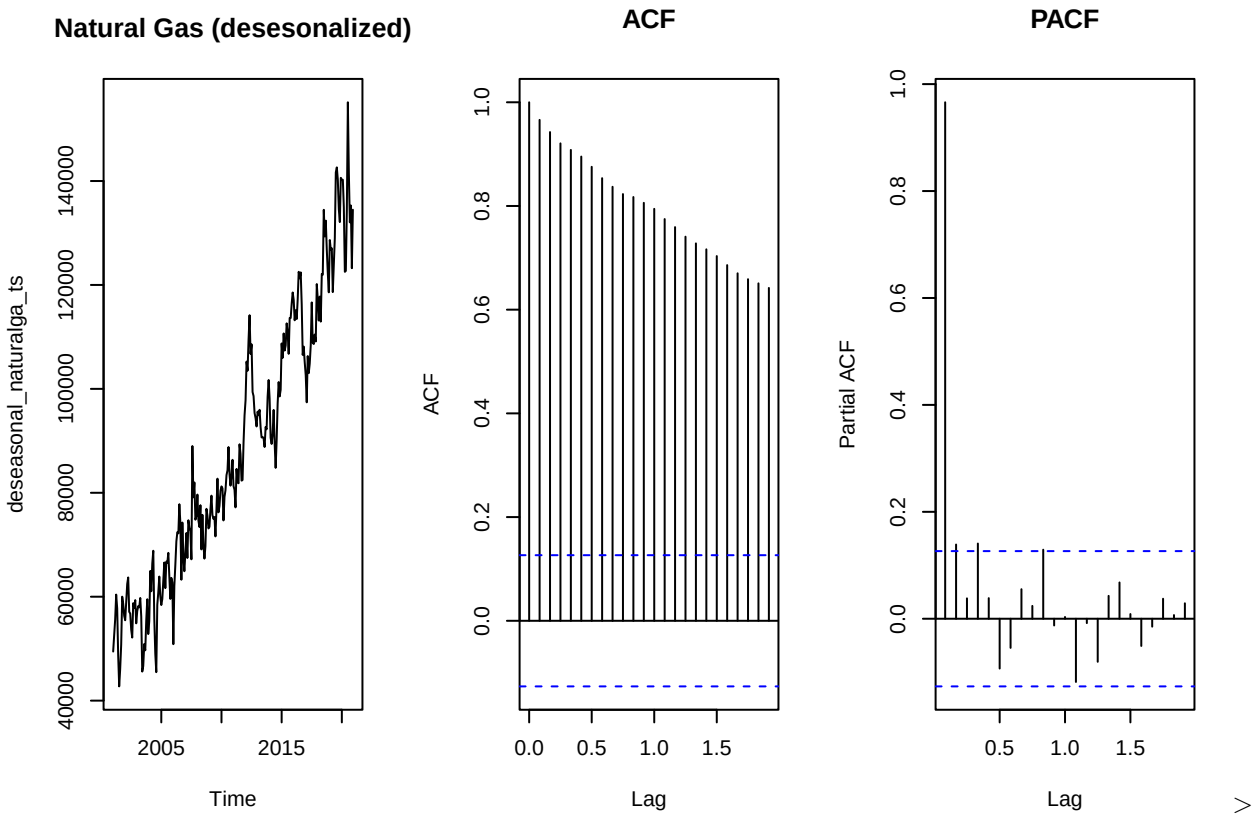


Q2

Using the *decompose*() or *stl*() and the *seasadj*() functions create a series without the seasonal component, i.e., a deseasonalized natural gas series. Plot the deseasonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

```
dc_naturalgas_ts <- decompose(naturalgas_ts)
deseasonal_naturalga_ts <- seasadj(dc_naturalgas_ts)

par(mfrow = c(1,3))
plot(deseasonal_naturalga_ts, main = "Natural Gas (desesonalized)")
acf(deseasonal_naturalga_ts, main = "ACF")
pacf(deseasonal_naturalga_ts, main = "PACF")
```



Answer: Compared to the data without seasonal adjustment, the time series with deseasonalised adjustment has smaller fluctuations and appears more stable. The ACF plot of the time series is also transformed from a curve shape with obvious seasonal fluctuations to a relatively smooth and decreasing line shape. The PACF plot only has a significant lag 1.

Modeling the seasonally adjusted or deseasonalized series

Q3

Run the ADF test and Mann Kendall test on the deseasonalized data from Q2. Report and explain the results.

```
#ADF
print("Results for ADF for deseasonalized data")
```

```
## [1] "Results for ADF for deseasonalized data"
```

```
print(adf.test(deseasonal_naturalga_ts,alternative = "stationary"))
```

```
## Warning in adf.test(deseasonal_naturalga_ts, alternative = "stationary"):
## p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  deseasonal_naturalga_ts
## Dickey-Fuller = -4.0271, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
#Mann Kendall
print("Results of Mann Kendall for deseasonalized data")
```

```
## [1] "Results of Mann Kendall for deseasonalized data"
```

```r
print(summary(MannKendall(deseasonal_naturalga_ts)))
```

```
## Score =  24186 , Var(Score) = 1545533
## denominator =  28680
## tau = 0.843, 2-sided pvalue =< 2.22e-16
## NULL
```

Answer: In ADF, the p-value is smaller than 0.05, which rejects the null hypothesis of a unit root and concludes that the deseasonalized natural gas series is stationary, no stochastic trend. In the Mann Kendall test, the 2-sided pvalue is smaller than 0.05, which rejects the null hypothesis and concludes that there have determinist trend.

Q4

Using the plots from Q2 and test results from Q3 identify the ARIMA model parameters p, d and q. Note that in this case because you removed the seasonal component prior to identifying the model you don't need to worry about seasonal component. Clearly state your criteria and any additional function in R you might use. DO NOT use the *auto.arima*() function. You will be evaluated on ability to can read the plots and interpret the test results.

```r
# whether d = 1 is enough
ndiffs(deseasonal_naturalga_ts, alpha = 0.05, test = c("kpss", "adf","pp"),max.d = 2)
```

```
## [1] 1
```

Answer: p,d,q = 1,1,0 1. Based on the ADF, the p-value is smaller than 0.05, indicating no stochastic trend. Meanwhile, the Mann-Kendall test shows there is determinist trend. 2. Due to ndiffs function result being 1, which means there only needs 1 differencing, therefore, d = 1. 3. In ACF, it's slow decay, and no cuts off. Therefore, it is an AR model, and q = 0. 4. In PACF, after lag1, there is a huge value decrease, although lag 2 is still significant, I still believe lag 1 is cut-off point. Therefore, p=1.

Q5

Use *Arima*() from package "forecast" to fit an ARIMA model to your series considering the order estimated in Q4. You should allow constants in the model, i.e., $include.mean = TRUE$ or $include.drift = TRUE$. **Print the coefficients** in your report. Hint: use the *cat*() function to print.

```r
ARIMA_deseasonal_naturalgas <- Arima(deseasonal_naturalga_ts, order = c(1,1,0), include.mean = TRUE)
#coefficients
cat("ARIMA coefficients:\n")
```

```
## ARIMA coefficients:
```

```
cat("Phi_1 = ", ARIMA_deseasonal_naturalgas$coef[1], "\n")
```

```
## Phi_1 =  -0.143964
```

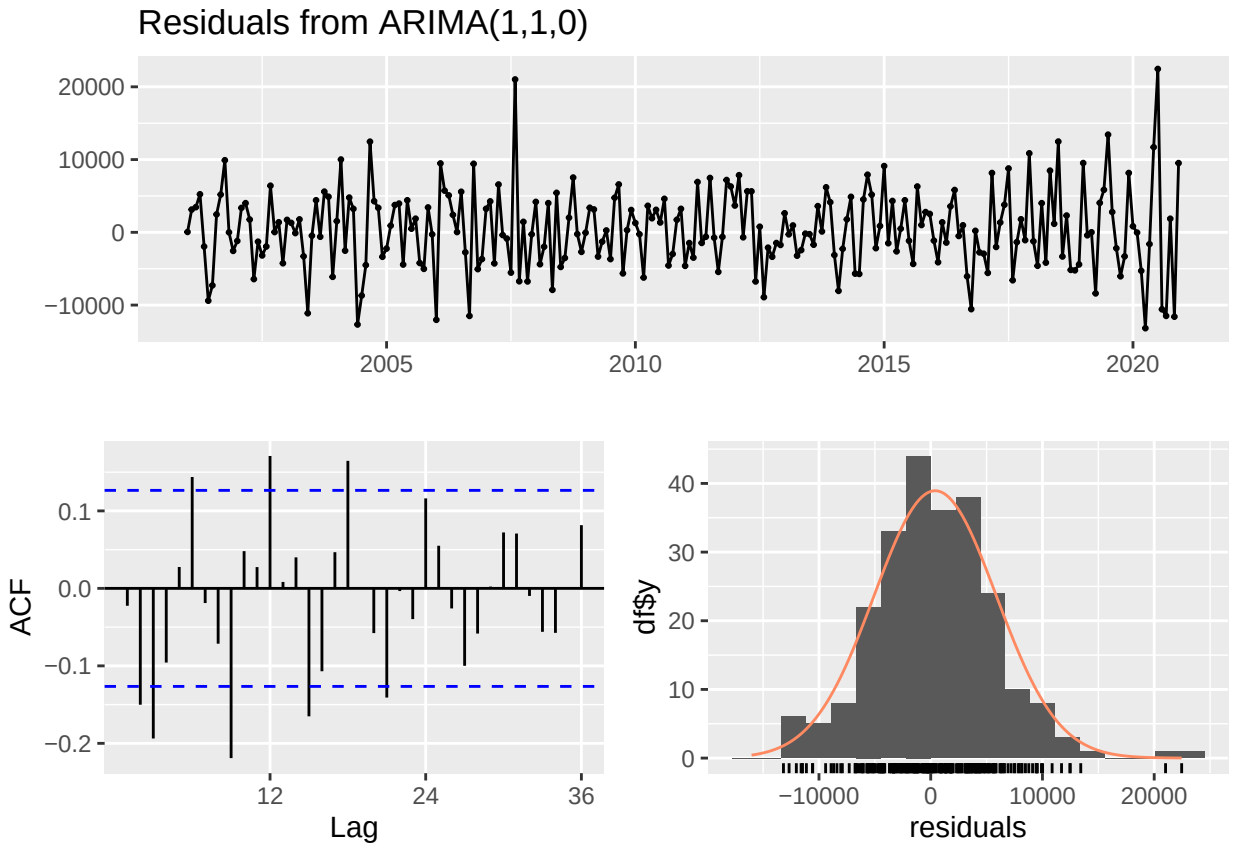
```
cat("Intercept = ", ARIMA_deseasonal_naturalgas$coef[2], "\n")
```

```
## Intercept =  NA
```

Q6

Now plot the residuals of the ARIMA fit from Q5 along with residuals ACF and PACF on the same window. You may use the *checkresiduals*() function to automatically generate the three plots. Do the residual series look like a white noise series? Why?

```
checkresiduals(ARIMA_deseasonal_naturalgas)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)
## Q* = 72.326, df = 23, p-value = 5.291e-07
##
## Model df: 1.   Total lags used: 24
```

Answer: The residual series does not look like a white noise series. The reason is that I observed significant autocorrelations at multiples of 3 in the ACF. And no (not too much) significant autocorrelations should exist at any lag in a white noise series.

Modeling the original series (with seasonality)

Q7

Repeat Q4-Q6 for the original series (the complete series that has the seasonal component). Note that when you model the seasonal series, you need to specify the seasonal part of the ARIMA model as well, i.e., P, D and Q.

```
#Q4,
##ADF
print("Results for ADF for original data")
```

```
## [1] "Results for ADF for original data"
```

```
print(adf.test(naturalgas_ts,alternative = "stationary"))
```

```
## Warning in adf.test(naturalgas_ts, alternative = "stationary"): p-value smaller
## than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  naturalgas_ts
## Dickey-Fuller = -8.9602, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
##Mann Kendall
print("Results of Mann Kendall for original data")
```

```
## [1] "Results of Mann Kendall for original data"
```

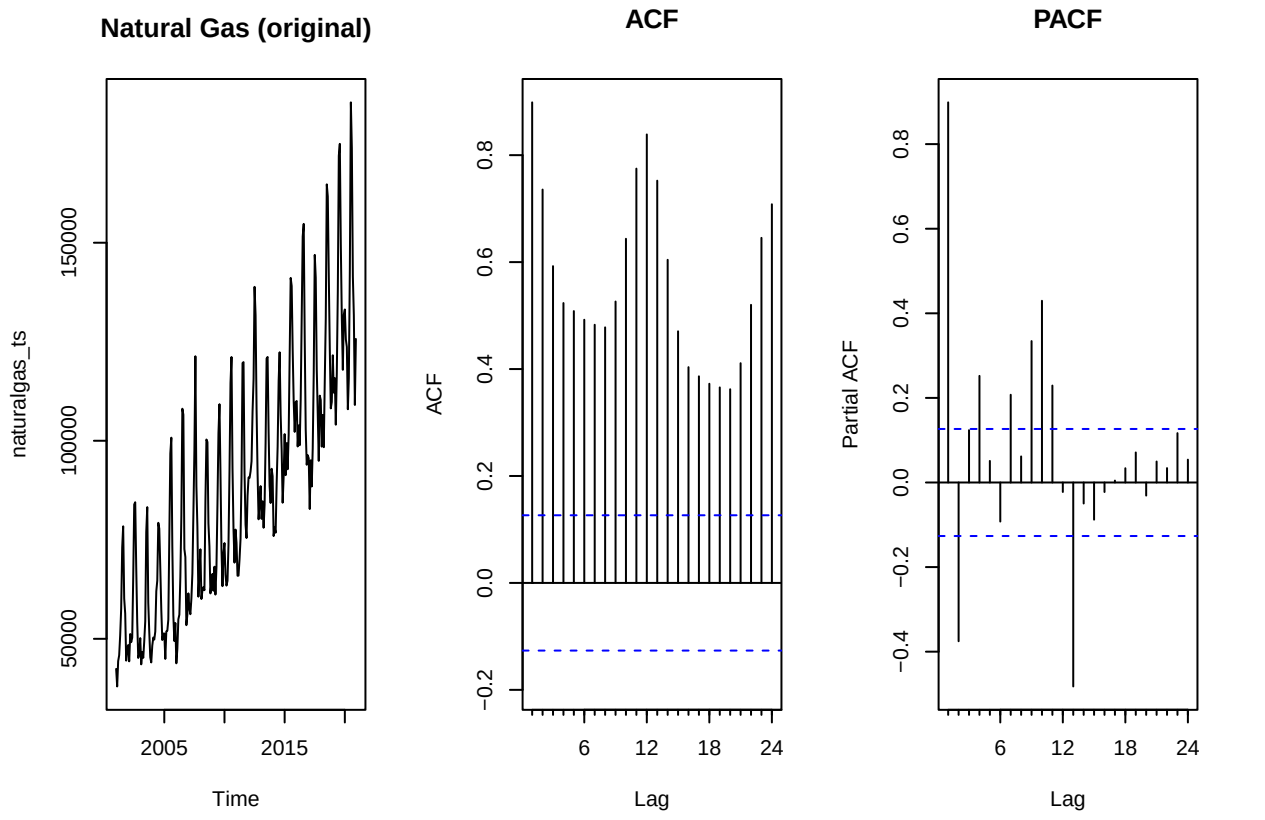
```
print(summary(MannKendall(naturalgas_ts)))
```

```
## Score =  18658 , Var(Score) = 1545533
## denominator =  28680
## tau = 0.651, 2-sided pvalue =< 2.22e-16
## NULL
```

```
## d
ndiffs(naturalgas_ts, alpha = 0.05, test = c("kpss", "adf","pp"),max.d = 2)
```

```
## [1] 1
```

```r
#PDQ,
par(mfrow = c(1,3))
plot(naturalgas_ts, main = "Natural Gas (original)")
Acf(naturalgas_ts, main = "ACF")
Pacf(naturalgas_ts, main = "PACF")
```




```r
##D
nsdiffs(naturalgas_ts, test = "ocsb", max.D = 1)
```

```
## [1] 0
```

```r
nsdiffs(naturalgas_ts, test = "ch", max.D = 1)
```

```
## [1] 0
```

```r
# SARIMA
SARIMA_naturalgas <- arima(naturalgas_ts, order = c(2, 1, 0), seasonal = list(order = c(1, 0, 0)))
##coefficients
cat("SARIMA coefficients:\n")
```

```
## SARIMA coefficients:
```

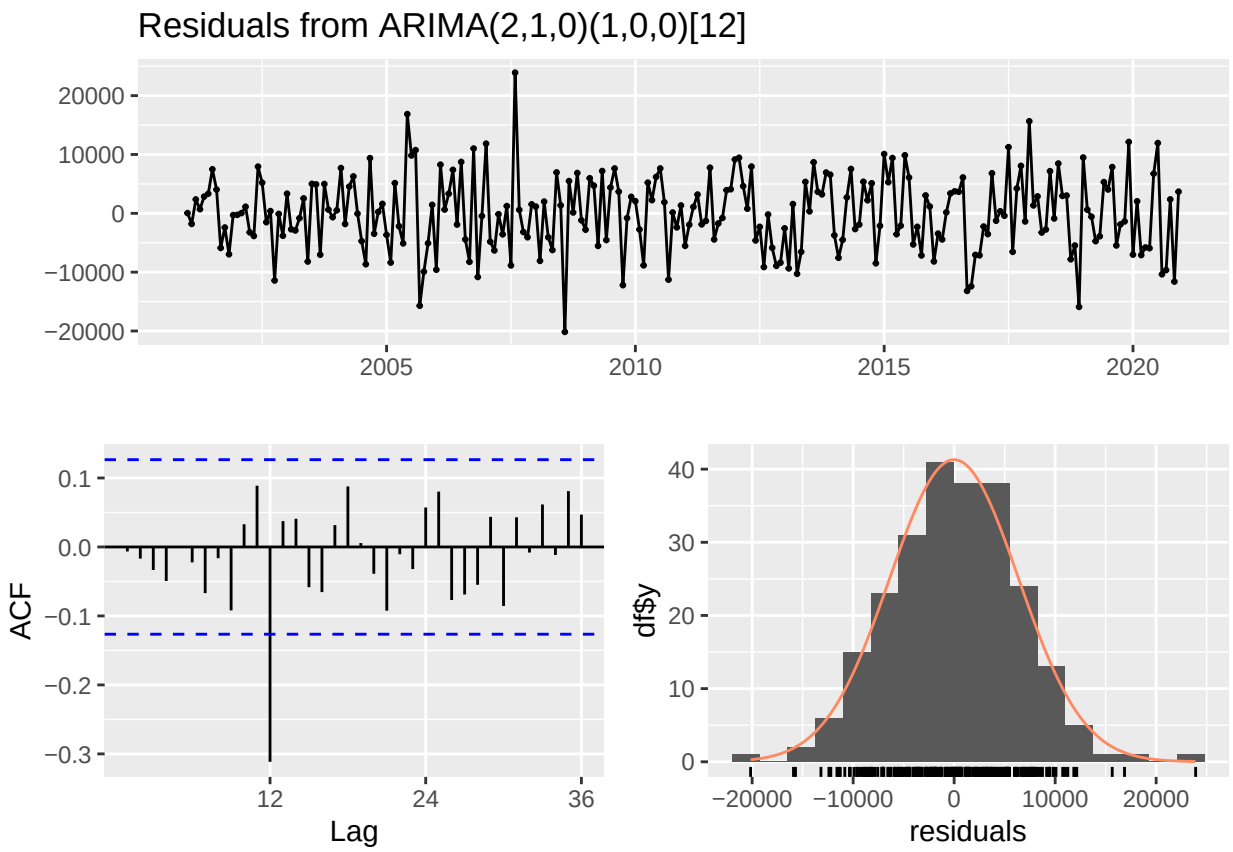
```r
cat("Phi_1 = ", SARIMA_naturalgas$coef[1], "\n")
```

```
## Phi_1 =  -0.1790847
```

```
cat("Intercept = ", SARIMA_naturalgas$coef[2], "\n")
```

```
## Intercept =  -0.2188623
```

```
#residuals
checkresiduals(SARIMA_naturalgas)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0)(1,0,0)[12]
## Q* = 40.27, df = 21, p-value = 0.006894
##
## Model df: 3.   Total lags used: 24
```

Answer: qdq, PDQ:(2,1,0),(1,0,0) 1. Based on the ADF and Mann-Kendall test, there is a determinist trend. 2. Due to ndiffs function result being 1, which means there only needs 1 differencing, therefore, d = 1. 3. In ACF, it's slow decay and no cuts off. Therefore, it is an AR model and q = 0 4. In PACF, after lag2, there is a huge value decrease. Although lag 3 is just a little smaller than the significant line and lag 4 is significant, I still believe lag 2 is the cut-off point. Therefore, p=2. 5. D =0 due to nsdiffs result is 0 6. In ACF, there are multiple spikes and the value for lag 1, 12, and 24, and their value is decreasing. Therefore it is SAR, and Q = 0. 7. In PACF, lag 1 and 2 are significant. Although lag 12 is not significant, lag 13 is the spike with a significant value. Therefore, I think the P = 1. Residual The residual series look like a white noise series. The ACF shows that only 1 lag's autocorrelation value is significant, and the distribution of this residual is close to a normal distribution with mean 0.

Q8

Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response.

```
SARIMA_naturalgas
```

```
##
## Call:
## arima(x = naturalgas_ts, order = c(2, 1, 0), seasonal = list(order = c(1, 0,
##     0)))
##
## Coefficients:
##           ar1      ar2    sar1
##       -0.1791  -0.2189  0.9067
## s.e.   0.0660   0.0636  0.0252
##
## sigma^2 estimated as 40771450:  log likelihood = -2443.6,  aic = 4895.2
```

```
ARIMA_deseasonal_naturalgas
```

```
## Series: deseasonal_naturalga_ts
## ARIMA(1,1,0)
##
## Coefficients:
##           ar1
##       -0.1440
## s.e.   0.0645
##
## sigma^2 = 30287053:  log likelihood = -2397.17
## AIC=4798.34   AICc=4798.39   BIC=4805.29
```

> Answer: Comparing residual series, the SARIMA model is better. If focused on the residual, SARIMA(2,1,0),(1,0,0) has adequately captured the underlying patterns in the data, and there is little structure remaining in the residuals. However, this result might be wrong due to the best way to evaluate the model is based on the predictive accuracy.

Checking your model with the auto.arima()

Please do not change your answers for Q4 and Q7 after you ran the *auto.arima*(). It is **ok** if you didn't get all orders correctly. You will not loose points for not having the same order as the *auto.arima*().

Q9

Use the *auto.arima*() command on the **deseasonalized series** to let R choose the model parameter for you. What's the order of the best ARIMA model? Does it match what you specified in Q4?

```
auto.arima(deseasonal_naturalga_ts)
```

```
## Series: deseasonal_naturalga_ts
## ARIMA(1,1,1) with drift
##
## Coefficients:
##          ar1      ma1     drift
##       0.7065  -0.9795  359.5052
## s.e.  0.0633   0.0326   29.5277
##
## sigma^2 = 26980609:  log likelihood = -2383.11
## AIC=4774.21   AICc=4774.38   BIC=4788.12
```

Answer: The result is ARIMA(1,1,1).It not match my specified (ARIMA(1,1,0))in Q4.

Q10

Use the *auto.arima*() command on the **original series** to let R choose the model parameters for you. Does it match what you specified in Q7?

```
auto.arima(naturalgas_ts)
```

```
## Series: naturalgas_ts
## ARIMA(1,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     sma1     drift
##       0.7416  -0.7026  358.7988
## s.e.  0.0442   0.0557   37.5875
##
## sigma^2 = 27569124:  log likelihood = -2279.54
## AIC=4567.08   AICc=4567.26   BIC=4580.8
```

Answer: The result is ARIMA(1,0,0)(0,1,1).It not match my specified (ARIMA(2,1,0)(1,0,0))in Q7.